LIN Requirements

COMMUNICATIONS SOFTWARE DEVELOPMENT IN EMBEDDED ENVIRONMENTS

Francisco Martinez Chavez

2-Feb-17

ITESO

Universidad Jesuita de Guadalajara

Prepared by: Francisco Martínez Chávez

Date: 2-Feb-17

Issue: 18

1. TABLE OF CONTENTS

2. LIN REQUIREMENTS

This document specifies the functionality, API and the configuration of the LIN driver module. The LIN driver applies to LIN specification 1.3 master nodes only.

3. LIN API SPECIFICATION

The specified interfaces and types defined next have to be fulfilled by the LIN driver.

3.1. IMPORTED TYPES

Here, all the other types imported from other modules are listed:

Module	*Imported Type*
Uart	Uart module types
Std_Types	Standard types, formerly named typedefs.h

3.2. TYPE DEFINITIONS

The content of Lin_Types.h consists of types specified within LIN Interface except for configuration types.

3.2.1. LINFRAMEPIDTYPE

Name	LinFramePidType	
Type	uint8_t	
Description	The LIN identifier (0..0x3F) along with its two parity bits	
Range	0..0xFE	Represents all valid protected identifier used by LinSendFrame()

3.2.2. LINFRAMECSMODELTYPE

Name	LinFrameCsModelType	
Type	Enumeration	
Description	Specifies the Checksum model used in the LIN Frame	
Range	LIN_ENHANCED_CS	Enhanced checksum model
	LIN_CLASSIC_CS	Classic checksum model

3.2.3. LINFRAMERESPONSETYPE

Name	LinFrameResponseType	
Type	Enumeration	
Description	Specifies whether the frame processor is required to transmit the response part of the LIN frame	
Range	LIN_MASTER_RESPONSE	Response is generated from this node (master)
	LIN_SLAVE_RESPONSE	Response is generated from a remote slave node

3.2.4. LINFRAMEDLTYPE

Name	LinFrameDlType

Type	uint8_t	
Description	Specifies the number of SDU of data bytes to copy	
Range	1..8	Data length of a LIN frame

3.2.5. LINPDUTYPE

Name	LinPduType	
Type	Structure	
Description	This type is used to provide PID, checksum model, data length and SDU pointer to the LIN driver	
Element	LinFramePidType	Pid
	LinFrameCsModelType	Cs
	LinFrameResponseType	Drc
	LinFrameDlType	Dl
	uint8_t*	SduPtr

3.3. FUNCTION DEFINITIONS

This is a list of functions provided for upper layer modules.

3.3.1. LIN_INIT

Service Name	Lin_Init	
Syntax	void Lin_Init (const LinConfigType* Config)	
Sync/Async	Synchronous	
Param (in)	Config	Pointer to Lin configuration
Param (out)	None	
Return value	None	
Description	Initializes the LIN module	

This function shall initialize the LIN module as well as the LIN channels. Note that different sets of configuration may be provided.

Initialization shall be according to the configuration set pointed by the parameter Config.

3.3.2. LIN_SENDFRAME

Service Name	Lin_SendFrame	
Syntax	Std_ReturnType Lin_SendFrame (uint8_t Channel, LinPduType* PduInfoPtr)	
Sync/Async	Asynchronous	
Param (in)	Channel	LIN Channel to be addressed
	PduInfoPtr	Pointer to PDU containing the PID, checksum, response, data length and SDU pointer
Param (out)	None	
Return value	Std_ReturnType	E_OK: Send command has been accepted
		E_NOK: Send command has not been accepted
Description	Sends a LIN Header and a LIN response if necessary according to the response type provided by the PduInfoPtr	

The function Lin_SendFrame shall send the header part (Break Field, Synch Byte Field and PID Field) and, depending on the direction of the frame response, a complete LIN response part of a LIN frames on the addressed LIN channel.

In case of receiving data the LIN Interface has to wait for the corresponding response part of the LIN frame by polling with the function Lin_GetSlaveResponse() after using the function Lin_SendFrame().

This function shall calculate the LIN Protected ID accordingly.

3.3.3. LIN_GETSLAVERESPONSE

Service Name	Lin_GetSlaveResponse	
Syntax	Std_ReturnType Lin_GetSlaveResponse (uint8_t Channel, uint8_t** LinSduPtr)	
Sync/Async	Synchronous	
Param (in)	Channel	LIN Channel to be checked
Param (out)	LinSduPtr	Pointer to pointer to a buffer mapped from the LIN module where the SDU is stored
Return value	Std_ReturnType	E_OK: SDU has been received and can be retrieved from the buffer
		E_NOK: SDU reception not complete yet
Description	Get the status of the LIN SDU reception	

This function shall return the current reception status of the LIN driver. If a SDU has been successfully received, the LinGetSlaveResponse shall store the SDU in a buffer referenced by LinSduPtr. The buffer will only be valid and shall be read until the next Lin_SendFrame function call.

4. DEPENDENCIES TO OTHER MODULES

Universal Asynchronous Receiver Transmitter

The hardware of the internal LIN driver depends on the internal MCU UART hardware. The LIN driver module will not take care of setting the registers that configure the clock, prescaler(s) and PLL (e.g. switching on/off the PLL) in its initialization functions. The UART driver module must do this as per expected configuration from the LIN driver module.

4.1. FILE STRUCTURE

4.1.1. CODE FILE STRUCTURE

The code file structure shall not be defined within this specification completely. At this point it shall be pointed out that the code-file structure shall include the following files named:

- Lin_Cfg.h – for definition configurable parameters, LIN configuration types and
- Lin_Cfg.c – for configurable parameters.

4.1.2. HEADER FILE STRUCTURE

The include file structure shall be as follows:

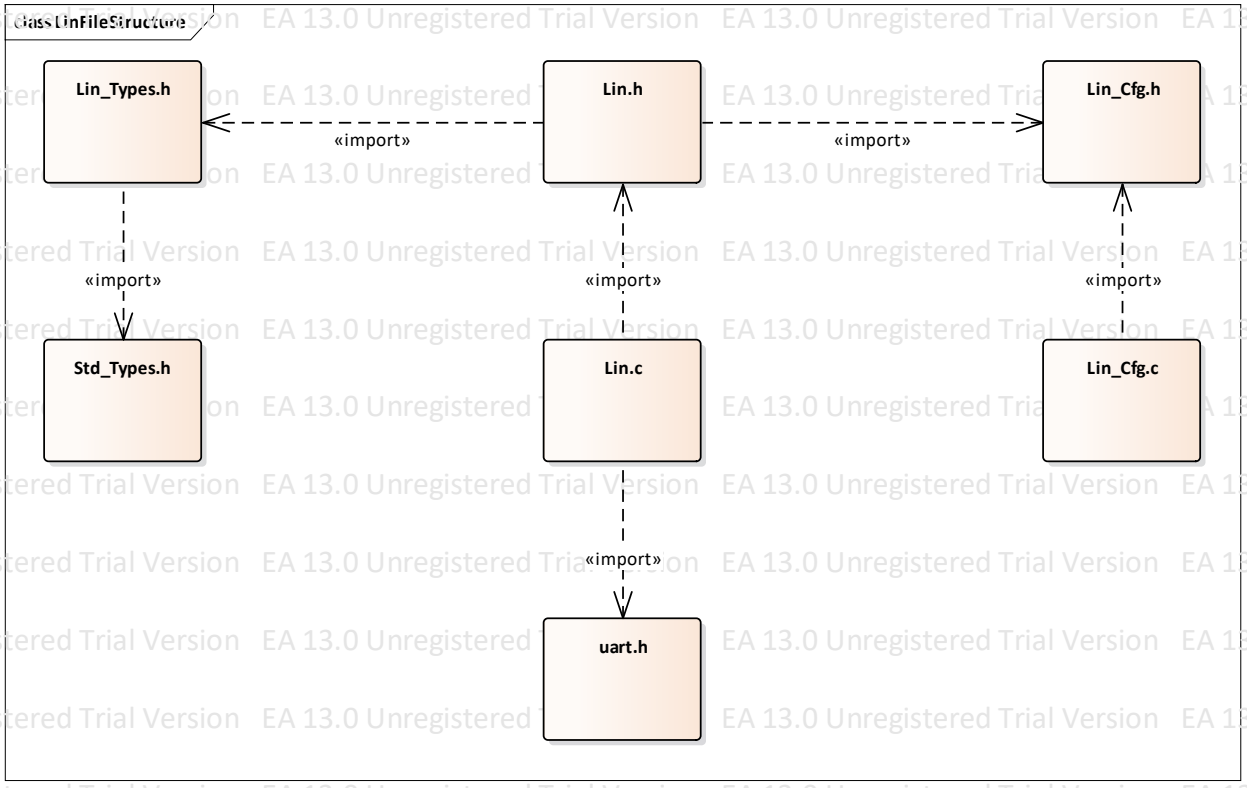

Figure 1: Header File Structure for the LIN driver

5. LIN CONFIGURATION SPECIFICATION

This chapter defines the configuration parameters and their structure (containers) of the module LIN driver.

5.1. CONTAINERS AND CONFIGURATION PARAMETERS

Configuration parameters define the variability of the generic part(s) of an implementation of a module. The main purpose is to provide a configurable module which can be adapted to the environment according to the target hardware and application in use.

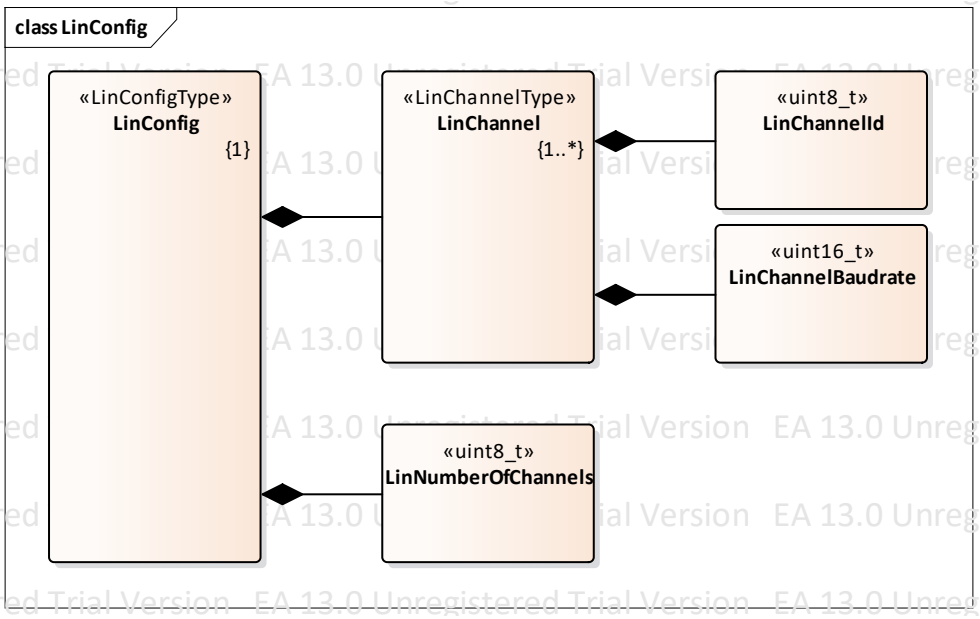

Figure 2: Configuration structure of the LIN driver

5.1.1. LINCONFIG

Name	*LinConfig*
Type	LinConfigType
Description	Configuration of the Lin (LIN driver) module

Included Containers		
Container Name	***Multiplicity***	***Description***
LinChannel	1..*	This container contains the parameters related to each LIN channel

Name	LinNumberOfChannels
Type	uint8_t
Description	Number of channels to be configured
Multiplicity	1
Range	1 .. 255

5.1.2. LINCHANNEL

Name	*LinChannel*
Type	LinChannelType
Description	This container contains the configuration parameters of the LIN channel

Not Included Containers

Name	LinChannelId
Type	uint8_t
Description	Lin Channel Identifier
Multiplicity	1
Range	1 .. 255

Name	LinChannelBaudrate
Type	uint16_t
Description	Specifies the baud rate of the LIN channel
Multiplicity	1
Range	1000 .. 20000